

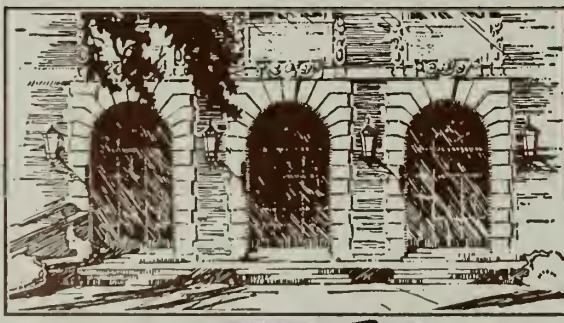
LIBRARY OF THE
UNIVERSITY OF ILLINOIS
AT URBANA-CHAMPAIGN

510.84

IL6r

no. 607-612

cop. 2



CENTRAL CIRCULATION AND BOOKSTACKS

The person borrowing this material is responsible for its renewal or return before the **Latest Date** stamped below. **You may be charged a minimum fee of \$75.00 for each non-returned or lost item.**

Theft, mutilation, or defacement of library materials can be causes for student disciplinary action. All materials owned by the University of Illinois Library are the property of the State of Illinois and are protected by Article 16B of *Illinois Criminal Law and Procedure*.

TO RENEW, CALL (217) 333-8400.

University of Illinois Library at Urbana-Champaign

APR 23 2004

When renewing by phone, write new due date
below previous due date.

L162

THE ICGS SYSTEM: SYSTEM TABLES MANUAL

by

David Henry Mueller

February 1974



DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

UIUCDCS-R-74-609

THE ICGS SYSTEM: SYSTEM TABLES MANUAL

by

David Henry Mueller

February 1974

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

* Supported in part by contract US AEC AT(11-1)2383

Table of Contents

Part I Introduction

1.	Introduction.....	1
1.1	Monitors and Overlays.....	1

Part II The System

2.	The Nucleus.....	4
2.1	The Scheduler (SVC).....	4
2.1.1	Line Printer Facility.....	4
2.1.2	Call for Next Monitor.....	4
2.2	The Overlay Facility (ØVRLAY).....	5
2.3	The Table Managing Facility (UPDATE).....	5
2.4	Other Facilities.....	6
3.	The Initializer.....	8
4.	Executing The System.....	8

Part III The Debugging Package

5.	The Overlay Builder (ØVRBLD).....	9
6.	The Debugging Initializer.....	10
6.1	The Input File.....	10
6.1.1	Record Description.....	10
6.1.2	File Description.....	12
	Bibliography.....	15



Digitized by the Internet Archive
in 2013

<http://archive.org/details/icgssystemsystem609muel>

ICGS System
Users Manual
Part I Introduction

1. Introduction

The ICGS System is a batch system for a mini-computer. It executes a limited number of system programs, and provides facilities for rapid overlaying of these programs.

The system was written for the following configuration:

- 1) PDP-11/20 with 16K of core memory
- 2) KE-11A Extended Arithmetic Element
- 3) Teletype
- 4) RF11 256K-2M word fixed-head disk
- 5) 2 TC11 Dectapes
- 6) GDI model 100 card reader
- 7) Gould 4800 printer/plotter

The System is intended to run under DOS. The System itself consists of a core resident Supervisor, a set of disk resident System modules, a set of desk resident user-defined modules, two bootstrap loaders, and an independent program to aid in the building of user modules. This Manual is intended to aid the user in preparing the modules and executing the system. Familiarity with the PDP-11 and DOS is assumed.

1.1 Monitors and Overlays

The Supervisor views the remainder of the system as a set of monitors, each of which may or may not possess a set of overlays. The set of overlays are structurally mutually exclusive, but the structural information for two overlays contained in the Supervisor may, in fact, refer to the same overlay file.

The Supervisor is responsible for reading in monitors and passing control to them. It is also responsible for reading in a monitor's overlays upon request and for storing the information necessary to do this.

The monitors are responsible for notifying the Supervisor of changes in the status of the overlays, calling the Supervisor to read in overlays, and directing the Supervisor what to do when the monitor has finished execution. Options for a monitor upon completion include running another monitor, turning to the batch input device for the next job, or returning to a previous monitor. The last option allows a monitor to temporarily relinquish control to another monitor, then regain control later.

All the facilities of DOS are also available to a given monitor. In effect the Supervisor and DOS combined appear to a monitor as a single operating system. Further, a monitor knows only of its own overlays; it may not have access to another monitor's overlays.

For purposes of batch execution, the monitors are grouped into sequences known as subsystems. The Grafix subsystem, for instance, consists of the Grafix compiler monitor and the Basic Plotting Package monitor. The structure of these subsystems is not contained in the Supervisor or its tables--it is defined by the monitors as each requests the next in the subsystem. The Supervisor maintains the subsystem name and the identity of the first monitor in each subsystem by which the user communicates with the system.

The Supervisor consists of the sequencer, the overlayer, and the table manager. There are two system monitors--one for using the Gould as a line printer, and one for handling system errors. The initializer is a bootstrap

routine which sets up the system tables and locates the monitors and overlays on the storage devices. Finally, there is a program to build overlay files and a version of the initializer specifically designed for debugging.

Part II The System

2. The Nucleus

2.1 The Scheduler (SVC)

The Scheduler controls the execution of the monitors. As each monitor returns to the Supervisor, it tells the Scheduler what is to be run next. The Scheduler then runs the line printer, if it has been requested to do so, determines the next monitor to be executed, reads it in, and gives it control. The call sequence is:

JMP SVC

2.1.1 Line Printer Facility

All line printer output from a monitor must be put in a single file on disk. Before exiting, a monitor which wants to run the line printer must put the radix 50 equivalent of the file name and extension in three consecutive words beginning at LPFILE and set an indicator byte LP to 1. Both LPFILE and LP are resident in the system and must be declared global. LP will be cleared after the line printer is executed.

2.1.2 Call for Next Monitor

Upon exit, a monitor must set a byte labeled NEXT to indicate what procedure is to be followed in determining the next monitor to be executed.

There are three options:

- 1) if NEXT is positive, the scheduler reads in the monitor whose number is given by NEXT.
- 2) if NEXT is negative, the monitor pushes the number of the monitor which just finished execution on a stack called the nest. It then takes the absolute value of NEXT to be the number of the next monitor to be executed.

- 3) if NEXT is zero, the scheduler takes the number of the next monitor to be executed from the top of the nest, and the nest. If the nest is empty, the Scheduler reads cards from the card reader until it reads a system control card (one with a '\$\$' in cc 1-2). It then uses the characters in cc 4-6 as a subsystem name and searches a list to find the monitor to execute. The card buffer is available with its line buffer header at the global location CRHEAD.

Possible errors in SVC include an attempt to execute a non-existent monitor, or to nest more than six monitors.

2.2 Overlay Facility (ØVRLAY)

The overlay facility may be called by a monitor or one of it overlays.

The overlay routine reads in an overlay and returns to the calling Program.

The call sequence is as follows:

```
JSR      R5, ØVRLAY
BR       .+4
.WORD ØVR
```

ØVR is the address of a byte containing the number of the overlay to be read in. A monitor may read in only its own overlays.

Possible errors include an invalid overlay number, an attempt to read in an uninitialized overlay (i.e. - one for which all parameters are not present in the system table), and an attempt to overlay the Supervisor.

2.3 The Table Managing Facility (UPDATE)

The Supervisor tables keep four parameters for each overlay:

- 1) The I/O unit on which the file containing the overlay is located.
- 2) The start block of the overlay file.
- 3) The start address of the overlay in core.
- 4) The length of the overlay, in words.

These parameters may be set by the initializer, and they must all be present

before ØVERLAY is called. If at any time one of the parameters changes, UPDATE must be called to make the change in the supervisor tables.

The call sequence is as follows:

JSR	R5,UPDATE
BR	CØNT
.BYTE	ØVERLAY
.BYTE	STATUS
other parameters	

CØNT:

ØVERLAY is the overlay number and STATUS is a status byte set up as in figure 7

1. The other parameters are:

- 1) File name and extension, in radix 50, present if bit 5 or 6 is 1
- 2) Memory start address, present if bit 3 is a 1
- 3) Length, in words, present if bit 4 is 1

These parameters are present if and only if the corresponding bits in the status byte are on. They must be in the order given, but any combination of them may be present.

A monitor may update only its own overlay table. If the routine fails, the result bit is set to 1 and no change is made to the tables.

2.4 Other Facilities

In addition to the routines above, the Supervisor provides the following facilities:

- 1) A link block for the disk labeled DFLINK, and one for the reader labeled CRLINK which have already been initied.
- 2) A link block for the teletype, labeled KBLINK, which has not been initied.

The Status Byte

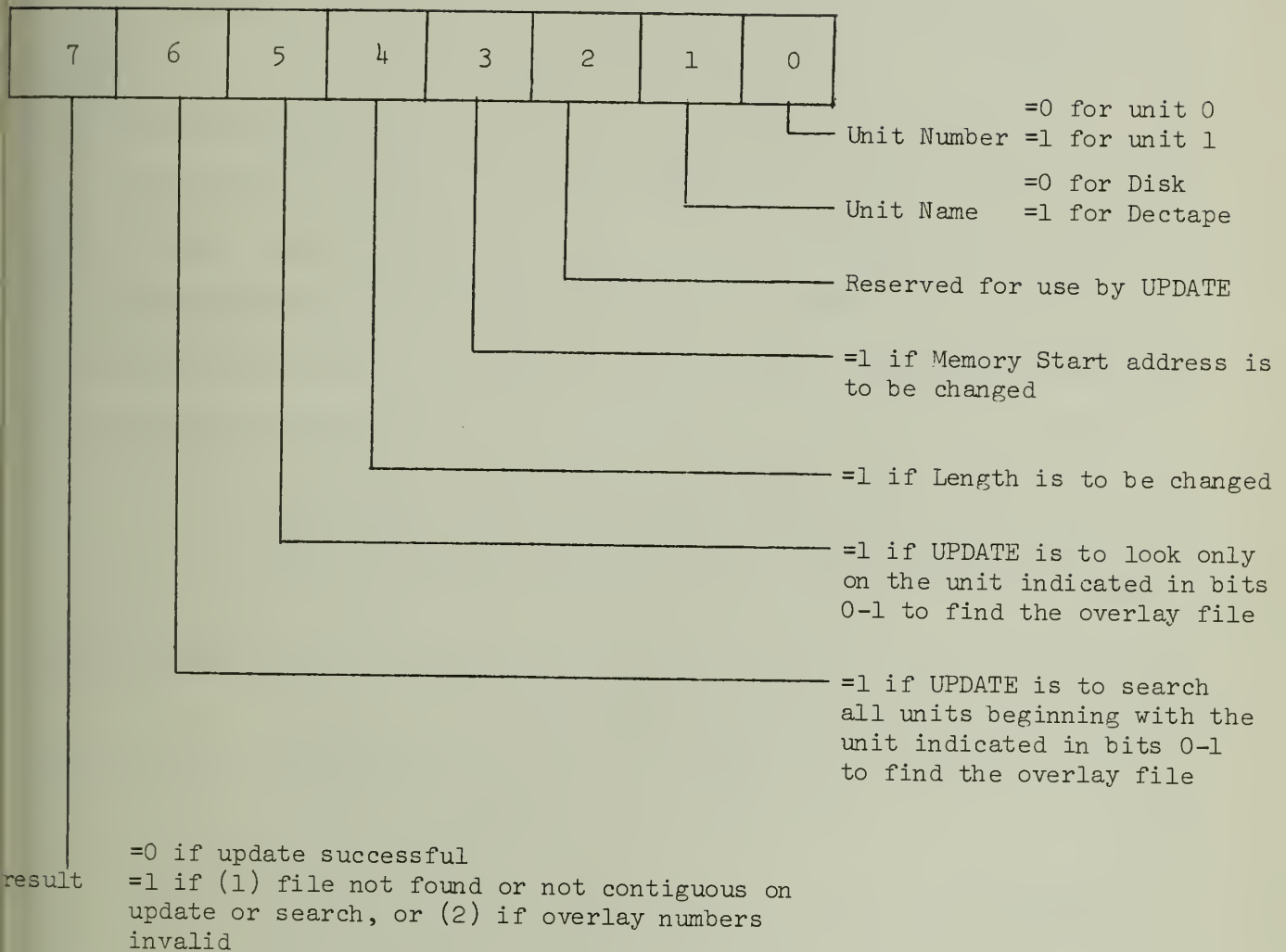


Figure 2.1

- 3) A ten-word field labeled PARAM, which can be used to pass information between monitors.
- 4) The system card read in by the monitor, with line buffer header CRHEAD.

3. The Initializer

The initializer is the first routine executed when the System is run. It initializes all system tables and variables, then transfers to SVC to begin execution.

The initialization of the System tables is controlled by tables within the initializer. The initializer tables consist of a table of transfer addresses for the monitors, a table giving the highest numbered overlay in each monitor, tables describing the subsystem, and, finally, a table giving the parameters for the overlays, i.e., the Status byte for use by UPDATE, the file name and extension of the overlay file, the memory start address, and the length of the overlay. All or none of these parameters will be initialized, depending on the status bytes.

A more detailed description of the initializer tables appears in the System Tables Manual.

4. Executing the System

Before executing the System, mount all DECTapes which are necessary. Log in, and type \$RUN SYSTEM. This will bring in the system nucleus and the initializer, run the initializer, and jump to the Scheduler, which will, in turn, read a control card from the card reader and execute the subsystem indicated.

5. The Overlay Builder (ØVRBLD)

The Overlay Builder is a program for building overlays to the system and monitors. It is a self-contained program which creates a contiguous file containing the overlay.

The overlay builder is executed by first running the Linker to obtain a load module, then executing the Overlay Builder to create the overlay. The load module which is created will contain only the modules in the overlay.

Run the Linker as is described in the Link-11 Linker Manual, with the following qualifications:

- 1) In addition to the output load module and load map, the load symbol table may be obtained by typing a file specification after the file specification for the load map. This will be used if further overlays are to be read in while this overlay is present.
- 2) The first input should be the load symbol table describing the remainder of the system configuration. This will always include the Supervisor nucleus, and may include ODT and other monitors and overlays. It will be obtained from a previous linkage. The symbol table for the Supervisor nucleus alone is SYSTEM.LSM, and for the nucleus and ODT, SYSTST.LSM.
- 3) After the last input file, on the first time specify the T switch as follows: /T:n, where n is the top word to be occupied by the overlay. Top addresses for the Supervisor are 075176 without ODT and 067434 with ODT.

After finishing with Link, type the following commands

\$GET ØVERLAY	where ØVRLAY is the load module
<u>\$</u> CTRL C	output from the Linker
.KI	
\$RUN ØVRBLD	
<u>#</u>	

This will load the overlay and the overlay builder into core. Then type a command string as follows:

file name specification/switch where switch is one of the following:

- 1) SL:s:l where s is the starting address of the overlay and l is its length in bytes.
- 2) SE:s:e where s is the starting address of the overlay and e is its ending address

The values of s, l, and e must be six digit octal numbers with leading zeros, if necessary. The overlay will be built under the name given in the file name specification.

6. The Debugging Initializer

This routine is similar to the regular initializer, but initialization of the monitor tables is controlled by an input file rather than core resident tables. These are described below. The intention is to give greater ease in altering the initial values of the tables.

6.1 The Input File

The input file consists of a series of records describing the monitors, overlays, and subsystems present. These are described below as cards, but they may be on any input device except the teletype keyboard. The records are described individually, then the make-up of the file is described.

6.1.1 Record Description

<u>Card ID</u>	<u>cc</u>	<u>Description</u>
\$1	1-2	Card ID-'\$1'-number of monitor card
	4-9	Number of monitors-minimum 1 maximum 10

<u>Card ID</u>	<u>cc</u>	<u>Description</u>
\$2	1-2	Card ID-'\$2'-Monitor Description Card
	4-9	Monitor Number ¹ -minimum 2
	11-13	Unit designation for the monitor file- DFO = Disk 0 DT0 = Dectape 0 DT1 = Dectape 1
	15-20	File Name of the monitor file ²
	22-24	Extension for the Monitor File ²
	26-31	Memory start address ³
	33	Switch-L = following field is length, in bytes E = following field is end address * = ignore following field
	35-40	Length, in bytes, or end address
	42-47	Transfer
	49-54	Number of overlays for this monitor
	56-61	Region size, in bytes, (i.e., the largest number of bytes in core occupied by this monitor its overlays at one time)
\$3	1-2	Card ID-'\$3'-overlay card
	4-9	Overlay number ¹ -minimum 0
	11-13	Unit of overlay file ² - DFO = Disk 0 DT0 = Dectape 0 DT1 = Dectape 1
	15-20	File name of overlay file ²
	22-24	Extension of overlay file ²
	26-31	Memory start address ³
	33	Switch-L = following field is length, in bytes E = following field is end address * = ignore following field

<u>Card ID</u>	<u>cc</u>	<u>Description</u>
\$4	1-2	Card ID-'\$4'-number of subsystem card
	4-9	Subsystem number ¹
	11-16	Monitor number of first monitor subsystem
	18-20	First three characters of subsystem name
\$0	1-2	Card ID-'\$0'-end of initialization

¹An asterisk (*) following this field will cause no action to be taken on the record.

²An asterisk (*) in the first character of the file name field will cause the file not to be located. The unit and extension fields will be ignored.

³An asterisk (*) in the first character of this field will cause the file to be ignored.

All numbers must be six digit octal numbers, with leading zeros, since the conversion routines will not accept leading blanks.

6.1.2 File Description

The first record must be a \$1 card. This is followed by sets of \$2 and \$3 cards, one set for each monitor. Each set consists of a \$2 card, followed by one \$3 card for each overlay to the monitor. There are no \$3 cards if the monitor has no overlays. After the last \$3 card comes a \$4 card, then one \$5 card for each subsystem. The file is terminated by a \$0 card.

There must be \$2 card for each monitor number from 2 to M+1, where M is the number in cc 4-9 of the \$1 card. For each \$2 card, there must be a set of \$3 cards, one for each overlay number between 0 and N-1, where N is the number in cc 49-54 of the corresponding \$2 card. All \$3 cards must be in sequence by overlay number behind the corresponding \$2 card and all \$2-\$3

card sets must be in sequence by the monitor number on the \$2 card. There must be one \$5 card for each subsystem number between 0 and S-1, where S is the number in cc 4-9 of the \$4 card. All \$5 cards must be in sequence by subsystem number. Finally, there may be no more than 24 \$2 and \$3 cards, that is, no more than 24 monitors and overlays.

6.2 Operating Procedures

The debugging version of the System consists of the debugging initializer ODT, and the Supervisor. To execute the debugging initializer, type the following commands.

- 1) \$GET SYSTST ODT - this brings debugging supervisor into core
- 2) \$ODT - this executes ODT
- optional 3) *076276;B - this will set a breakpoint on the last instruction of the Scheduler. When this point is reached, the monitor is in core and may be debugged.
- optional 4) *076764;B - This will set a breakpoint on the last instruction of ØVRLAY. When this point is reached, the overlay is in core and may be debugged.
- 5) *063474;G - this begins execution of the supervisor initializer
- 6) #file name specifcation/OD - This gives the input file for the debugging initializer.

The initializer will finish execution and transfer to SVC.

It is the responsibility of the monitor to keep track of which overlays are in core and at what points another overlay is required. Since the Supervisor cannot monitor I/O operations, the monitor itself must also notify the Supervisor when the location of an overlay file has changed. A monitor

knows the location of its file by the file name, the Supervisor, upon request, look this file up in the directory of one of the mass storage devices and records the physical location of the file in its tables. The Supervisor can then use this information to read the file in rapidly upon request.

Bibliography

- Brown, R. Leonard, Jr., Numerical Systems on A Minicomputer, Report No. UIUCDCS-R-73-555 AEC-COO-1469-0215, Department of Computer Science, University of Illinois, Urbana, February, 1973.
- Digital Equipment Corporation, PDP 11-20 Processor Handbook, Maynard Massachusetts, 1971.
- Mueller, David, A Batch System with Rapid Overlay Capabilities, Report No. UIUCDCS-R-73-614 AEC-COO-2383-0005, Department of Computer Science, University of Illinois, Urbana, January, 1974.
- Mueller, David, The ICGS System: System Tables Manual, Report No. UIUCDCS-R-73-609 AEC-COO-2383-0003, Department of Computer Science, University of Illinois, Urbana, January, 1974.

U. S. ATOMIC ENERGY COMMISSION
UNIVERSITY-TYPE CONTRACTOR'S RECOMMENDATION FOR
DISPOSITION OF SCIENTIFIC AND TECHNICAL DOCUMENT

(See Instructions on Reverse Side)

1. AEC REPORT NO.

AEC-COO-2383--0002

2. TITLE

THE ICGS SYSTEM: SYSTEM TABLES MANUAL

TYPE OF DOCUMENT (Check one):

- ☒ a. Scientific and technical report
☐ b. Conference paper not to be published in a journal:

Title of conference _____

Date of conference _____

Exact location of conference _____

Sponsoring organization _____

- ☐ c. Other (Specify) _____

RECOMMENDED ANNOUNCEMENT AND DISTRIBUTION (Check one):

- ☒ a. AEC's normal announcement and distribution procedures may be followed.
☐ b. Make available only within AEC and to AEC contractors and other U.S. Government agencies and their contractors.
☐ c. Make no announcement or distribution.

REASON FOR RECOMMENDED RESTRICTIONS:

SUBMITTED BY: NAME AND POSITION (Please print or type)

C. W. Gear

Professor and Principal Investigator

Organization

Department of Computer Science
University of Illinois
Urbana, Illinois 61801

Signature

Charles W. Gear

Date

February 1974

FOR AEC USE ONLY

AEC CONTRACT ADMINISTRATOR'S COMMENTS, IF ANY, ON ABOVE ANNOUNCEMENT AND DISTRIBUTION
RECOMMENDATION:

PATENT CLEARANCE:

- ☐ a. AEC patent clearance has been granted by responsible AEC patent group.
☐ b. Report has been sent to responsible AEC patent group for clearance.
☐ c. Patent clearance not required.

BIBLIOGRAPHIC DATA SHEET	1. Report No.	UIUCDCS-R-74-609	2.	3. Recipient's Accession No.
	Title and Subtitle			5. Report Date
THE ICGS SYSTEM: SYSTEM TABLES MANUAL			6.	February 1974
Author(s)			8. Performing Organization Rept. No.	
David Henry Mueller				
Performing Organization Name and Address			10. Project/Task/Work Unit No.	
Department of Computer Science			11. Contract/Grant No.	
University of Illinois			US AEC AT(11-1)2383	
Urbana, Illinois 61801			13. Type of Report & Period Covered	
Sponsoring Organization Name and Address			Thesis Research	
US AEC Chicago Operations Office			14.	
9800 South Cass Avenue				
Argonne, Illinois				
6. Supplementary Notes				
7. Abstracts				
<p>The Illinois Computing Graphics System (ICGS) is a batch system for a minicomputer. It executes a limited number of system programs, and provides facilities for rapid overlaying of these programs. This manual is intended to aid the system programmer in setting up the system tables.</p>				
7. Key Words and Document Analysis. 17a. Descriptors				
<p>Supervisory Systems Executive routines overlays minicomputers</p>				
7b. Identifiers/Open-Ended Terms				
7c. COSATI Field/Group				
8. Availability Statement			19. Security Class (This Report)	21. No. of Pages
unlimited			UNCLASSIFIED	27
			20. Security Class (This Page)	22. Price
			UNCLASSIFIED	

JUL 18 1924



UNIVERSITY OF ILLINOIS-URBANA



3 0112 064441527